

## Recitation 10

### Misc programming skills and help with WA4

---

## 1 Java Doc

javaDoc is a special kind of comment in java code that can be used to generate documentation automatically. javaDoc comments have `/* ...*/` form, and look for keywords that begin with the `@` sign, such as `@author`, `@param`, `@return`. javaDoc comments go just before a class or method. javaDoc generates HTML documentation: much of the java documentation on the Oracle site was generated using it. You should put javaDoc comments on all your classes and methods for this course.

## 2 Subversion

Subversion is a source control tool. That means it is a tool that allows multiple users to share a single code base. The model is that there is a remote code repository that a group shares. There are also local copies that individuals work on. When you're ready to work, you copy the code from the repository, do some work, then put your changes back into the repository. If two people modify the same file, SVN tries to be smart about merging it together. If you're modifying code that overlaps, that is a conflict, and in that case SVN may need some help to merge the changes. That's advanced, and we'll just cover the basics today.

Subversion keeps track of all changes that are checked in, and keeps a copy of every version of every file that's checked in.

Getting the code has two parts. To start working with a code base, you "checkout". Each day, to get whatever changes have been made to the code base you "update". To send your changes to the code base you "checkin". This can all be done from the command line, but we're going to talk about how we can do as much as possible from within Eclipse.

There are two tools to use within Eclipse to connect to Subversion. We'll discuss subclipse, not subversive. If you use Windows, you may want to look into the TortoiseSVN client – it's not within Eclipse, but it's very easy to use. We won't talk about TortoiseSVN.

There are 3 main steps to set up within Eclipse:

1. Install Subclipse
2. Create a repository
3. Connect an Eclipse project to the repository.

Step 1 is done by each partner once. Step 2 is done by one partner each time you want to create a new repository (perhaps once for each assignment). Step 3 is done by both partners once for each project or repository.

For today's practice, we don't have partners yet, so each person can do all steps to set up a private subversion repository for practice.

## 2.1 Installing Subclipse

1. Start Eclipse
2. Choose Help > Install New Software
3. Type `http://subclipse.tigris.org/update_1.10.x` into “work with”
4. Check the following items:
  - Core SVN Kit Library, Optional JNA Library, Subclipse,
  - Subversion Client Adapter, and Subversion JavaHL Native ...
5. Then follow the prompts to install it all, accepting the license, etc.

## 2.2 Creating a Subversion Repository

Do this step each time you start a new project.

1. Open a terminal window
2. Find the absolute location of your group directory:  
`cd ~/cs200/cs200a` *today, just cd to your home directory*  
`pwd`
3. Make a note of the result. It will be something like this:  
`/s/bach/a/class/cs200/cs200a` *today, it's probably something like /s/chopin/a/under/zzz*
4. Create a folder to hold your repositories:  
`mkdir SVNREPO`
5. Create a repository (put your group name in for ###)  
`svnadmin create /s/bach/a/class/cs200/###/SVNREPO/`  
*today, use what the pwd command told you earlier*
6. Now, set up the correct permissions in the SVNREPO. Put your group name in for ### and don't miss the periods. *today and for personal repositories skip this step*  
`cd SVNREPO`  
`chgrp -R ### .`  
`chmod g+rws .`  
`find . -type f -print | xargs chmod g+rw`  
`find . -type d -print | xargs chmod g+rwxs`

## 2.3 Connect an Eclipse project to the repository

1. Create a new java project (just like always)
2. Right-click it, and choose Team > Share Project
3. Choose SVN, not CVS (CVS is another source control tool)
4. Create a new repository location:  
`file:///s/bach/a/class/cs200/###/SVNREPO/` *today, file:///s/chopin/a/under/zzz*
5. Follow the prompts. If someone else created the project first, you will be connected to the existing project.

## 2.4 Usage

When about to work, start by doing right-click, Team > Update to HEAD Then do your work. When done, do right-click, Team > Commit

You can commit more than once in a session to save your state. You can also try out some of the other commands in the Team menu. If you need more information, you can look at

- <http://svnbook.red-bean.com/> for docs on Subversion
- <http://subclipse.tigris.org/documentation.html> for Subclipse

## 2.5 Remote usage

This can be tricky, and we won't be covering this in detail today. We're just going to give you some pointers you can use to figure it out on your own. You will need to set up your machine to use ssh. You want your ssh keys to be set up, and you want to be authenticated with ssh before you start Eclipse. To set up your ssh keys, you can take a look at:

<http://pkeck.myweb.uga.edu/ssh/>

You may also need the following in your .profile file if you're using a macintosh remotely:

```
export SVN_SSH='ssh -l $USER'
```

Then, when you try to connect to the repository, your url will look like this:

```
svn+ssh://user@machine.cs.colostate.edu/s/bach/a/class/cs200/###/REPO/
```

Substitute in your username and your favorite machine.